# -\*- coding: utf-8 -\*-

"""

Created on Thu Jun 28 16:51:21 2018

@author: shubham b thorat

"""

# importing the libraries

"""

university ranking prediction

"""

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

# importing the data set

dataset=pd.read\_csv('university\_ranking.csv')

X = dataset.iloc[:,1:13].values

Y = dataset.iloc[:,0].values

# taking care of missing data

from sklearn.preprocessing import Imputer

imputer = Imputer(missing\_values="NaN", strategy="mean",axis=0)

imputer = imputer.fit(X[:,2:13], y=None)

X[:,2:14] = imputer.transform(X[:,2:14])

# handling categorical (encoding) data

from sklearn.preprocessing import LabelEncoder

label\_encoder\_X = LabelEncoder()

label\_encoder\_Y = LabelEncoder()

X[:,0] = label\_encoder\_X.fit\_transform(X[:,0])

X[:,1] = label\_encoder\_X.fit\_transform(X[:,1])

#onehotencoder = OneHotEncoder(categorical\_features= [0])

#onehotencoder = OneHotEncoder(categorical\_features= [1])

#X = onehotencoder.fit\_transform(X).toarray()

# dividing dataset into test and training dataset

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X,Y, test\_size = .3, random\_state = 0)

# feature scaling

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.fit\_transform(X\_test)

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

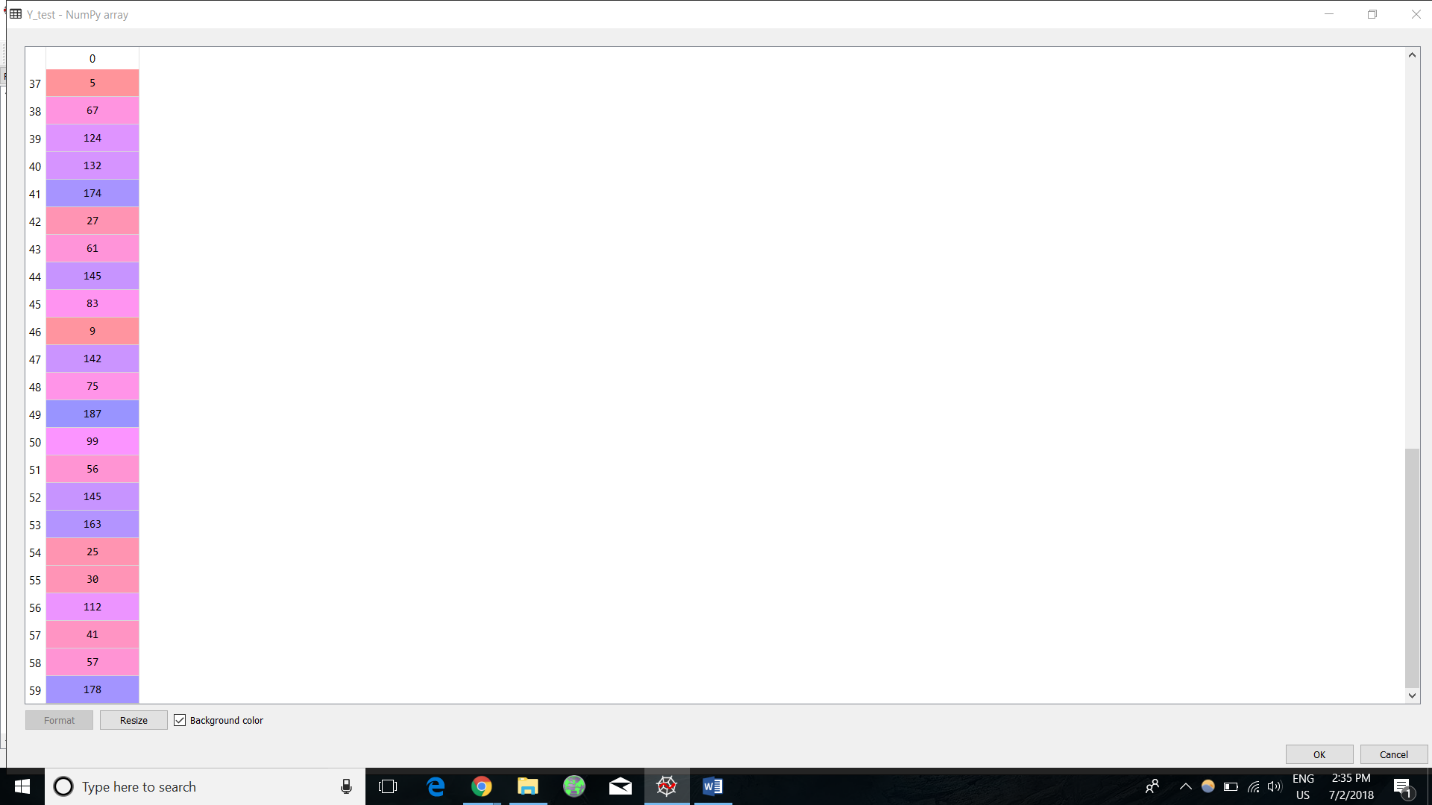
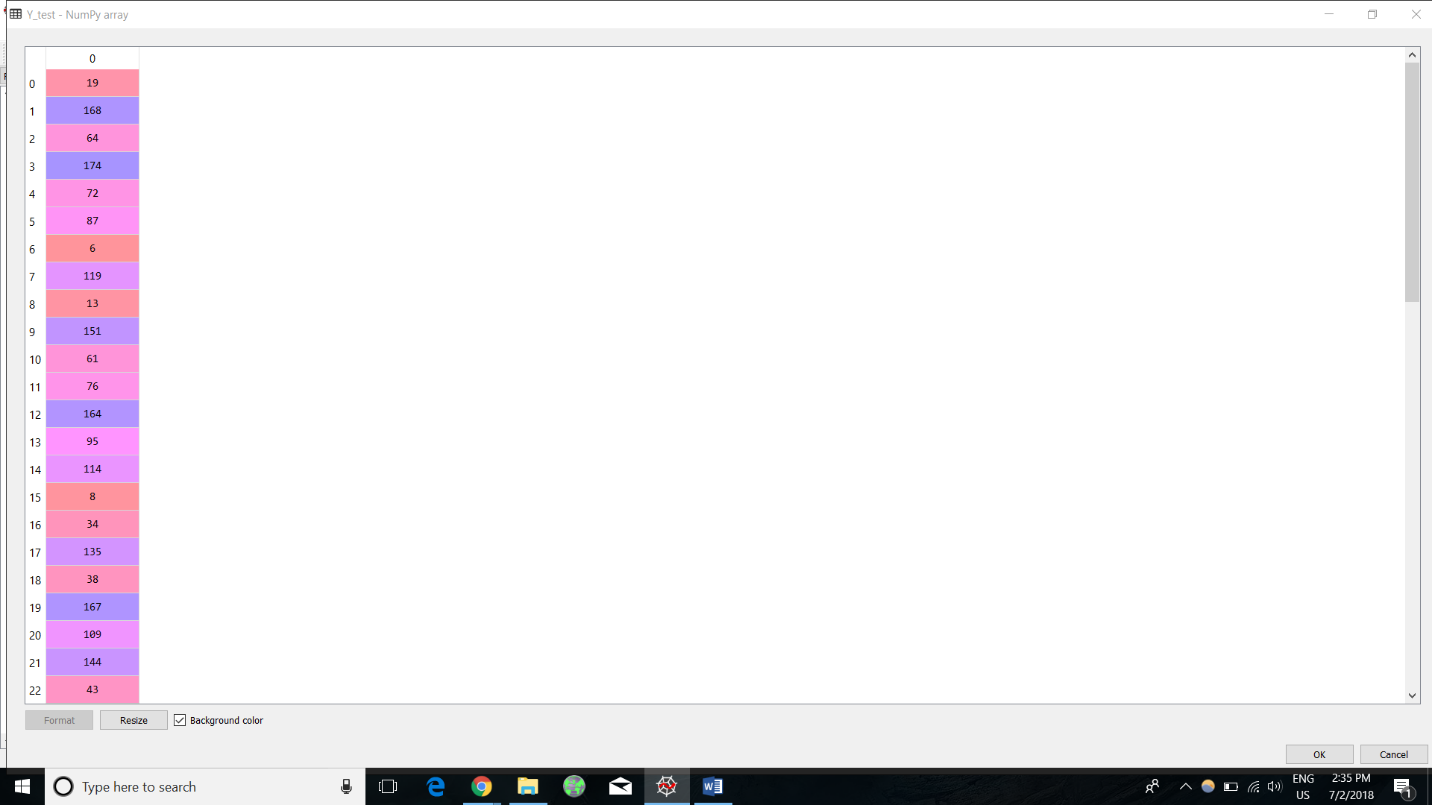
regressor.fit(X\_train,Y\_train)

#predicting the test set result

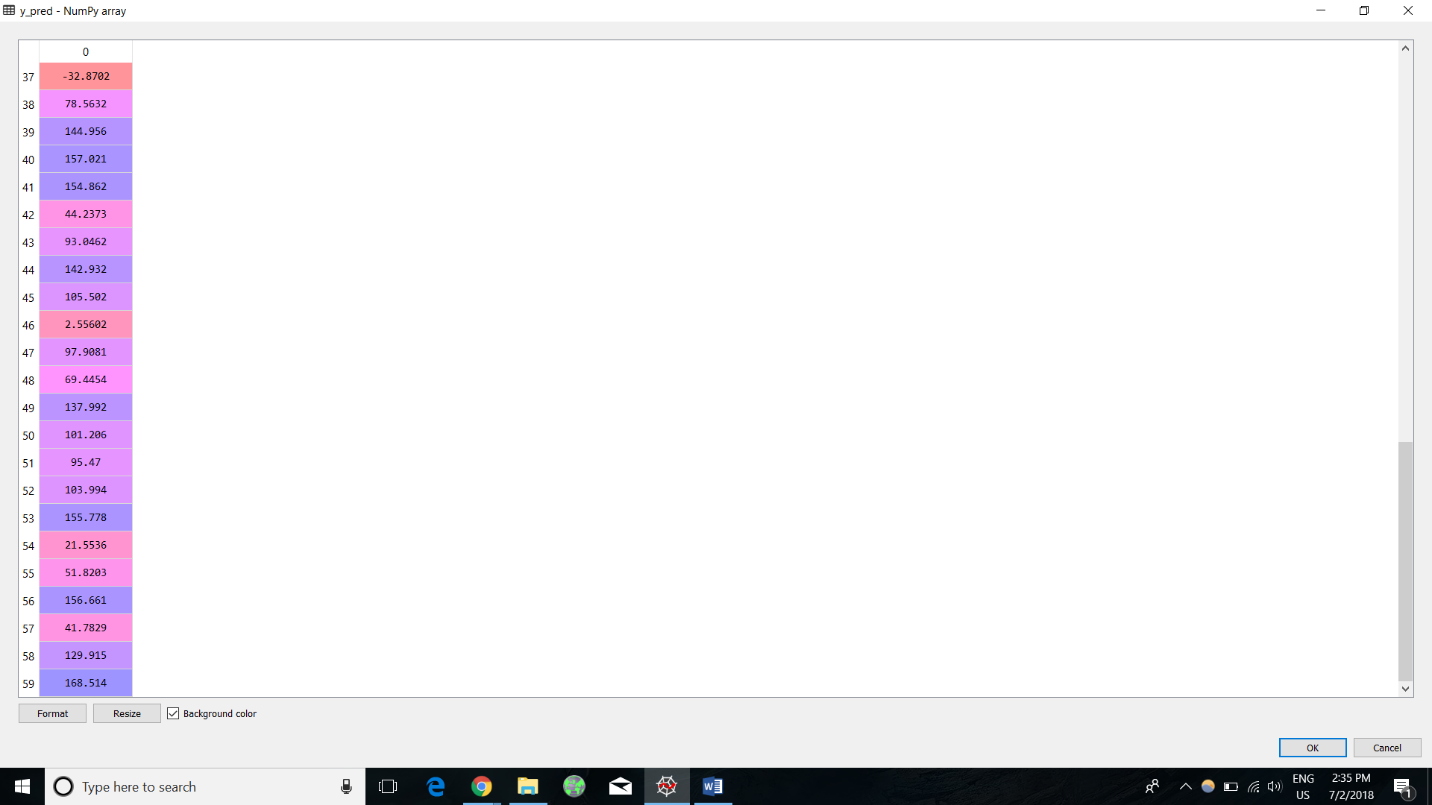
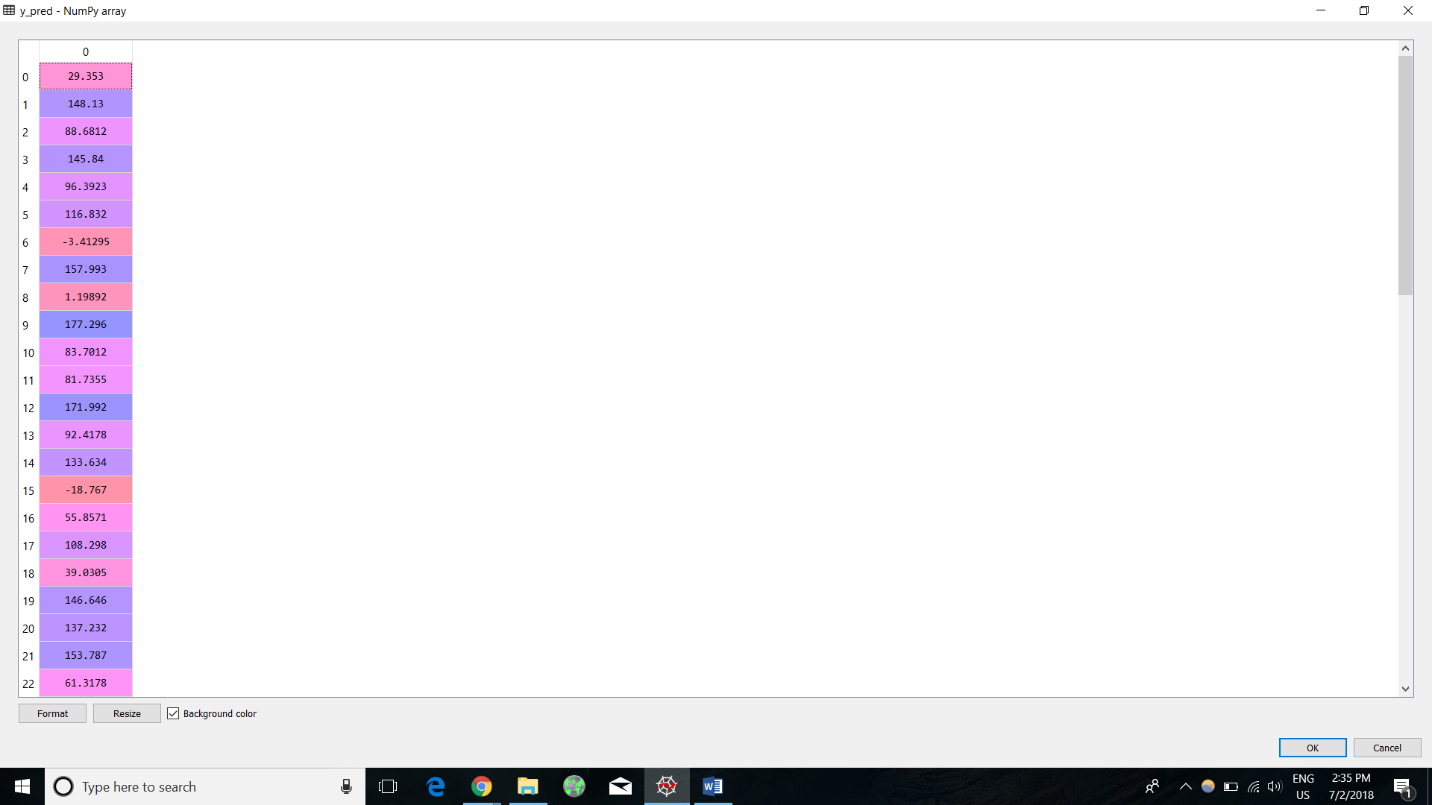
y\_pred = regressor.predict(X\_test)

“””

Y\_test =



Y\_pred =



“””